Register File Design Optimization with Virtual Prototyping (ViPro) Tool

Ningxi Liu[1], Shuoshuo Chen[2]
ECE 6332
University of Virginia
nl6cg@virginia.edu[1],
sc7cq@virginia.edu[2]

ABSTRACT

In this paper, we describe our work of applying virtual prototyping (ViPro) tool into register file design optimization. ViPro realizes fast and accurate simulation which can reduce the simulation time by 10 times and only has around 10 percent discrepancy in results compared with real register file netlist. According to our analysis assisted by ViPro, the optimal bank structure with minimum delay occurs when number of column mux equals 1, without considering the area overhead; the optimal bank structure with minimum energy consumption occurs when row number is about 64. Furthermore, the benefits of employing a sense amplifier in register file are only significant in larger memory capacity, due to the remarkable BL differential setup delay resulted from big row number.

1. INTRODUCTION

Register file is an array of processor registers in a central processing unit, and it usually has multiple read and write ports aimed at fast operation. Register file takes up a significant fraction of the power budget of processors, and it contributes to more than 25% of total power consumption in low power applications, according to Nalluri's work (2007) [1]. As a result, finding the optimized trade-off point between delay and energy consumption is critical in register file design. In addition, it's common that more than 30 unique and custom register files are employed in a single CPU and Soc chip (Eric Donkoh, DAC2012) [2]. Different structures of register file bank lead to different delay and energy, while full custom design for each case requires huge amount of time and effort. ViPro provides a good opportunity to accelerate register file design and optimization, since it can rapidly evaluate different bank structures without fully building sub-circuits [3] [4].

A work on cache design named CACTI from HP labs functions similarly as ViPro does. Compared with existing CACTI, ViPro gives simulation results based on SPICE model and real technology features, while CACTI uses a pure mathematical model to predict, which has low precision. So ViPro has the potential to outperform CACTI with a more accurate and real technology related model.

In this paper, comparison between ViPro and real register file schematic is proposed in chapter 2 to verify the simulation accuracy of ViPro. Chapter 3 includes the detailed delay model which can generate separated components of read and write delay. In this chapter, the optimal bank structure is researched across different memory size. Chapter 4 mainly answers the question that what's the influence on register file with or without using a sense amplifier.

2. VIPRO'S ACCURACY VERIFICATION

ViPro is capable of evaluating the trade-off between delay and energy in micro-architecture level by an abstract register file model. According to [3] and [4], ViPro is very efficient in simulation because its model is consisted of small components of a register file bank, which makes the model to be extremely concise. However, we can't get the benefits of efficiency without losing accuracy, and the results of ViPro are valid only if its model is correct and accurate enough. The best way of verifying ViPro is to compare its results with those generated from a real register file schematic. To implement this idea, four cases of real register file bank schematic were constructed with SKILL script, which represented four bank prototypes that could be evaluated by ViPro. We chose four characterizations from both ViPro and register file schematic as the standards for comparison, and they were read delay, read energy, write delay and write energy. If the differences of those results are not significant, then ViPro could be proved to be accurate; otherwise, a more precise model should substitute the existing one.

Table 1 Delay & energy of read & write operation in ViPro and real register file schematic

Case	Type	Delay(ns)		Energy(pJ)	
		Read	Write	Read	Write
1	ViPro	1.45	0.8	0.75	0.72
	Sch.	1.43	0.74	0.73	0.72
2	ViPro	1.46	0.8	0.81	0.75
	Sch.	1.47	0.76	0.79	0.73
3	ViPro	1.49	0.81	0.94	0.8
	Sch.	1.5	0.81	0.88	0.79
4	ViPro	1.53	0.84	1.19	0.9
	Sch.	1.6	0.89	1.06	0.89

The comparison results are shown in Table 1. All the cases were simulated under row number of 64 and wordsize of 16; the numbers of column sharing a sense amplifier are 1, 2, 4 and 8 for case 1-4. Figure 1 illustrates the percentage of differences about energy and delay's simulation results between Vipro and register file schematic. According to the figure, except for the read energy in case 4, the differences are all less than 10 percent, and a large portion of them is even approaching 0.

As a conclusion, the result implies prominent accuracy of ViPro in consideration that its speed of simulation is 10x faster than the real register file schematic.

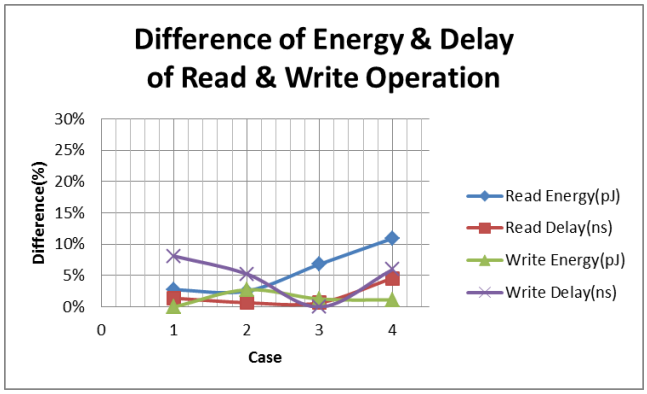

Figure 1. Differences of energy & delay of read & write operation between Vipro and register file schematic

3. OPTIMIZING BANK STRUCTURE OF REGISTER FILE

A proper register file design depends on specifications, such as memory size, delay and energy consumption, according to different kinds of applications. Within a given memory size, ViPro is a powerful tool that can rapidly generating various structures of register file prototype with energy and delay features. We can easily choose an optimized bank structure in requirements on delay and energy. Otherwise, some assist techniques should be used to realize more competitive register file banks. In addition, we are also interested in the optimized bank structure when memory size is changing.

3.1 Components of Delay and Energy

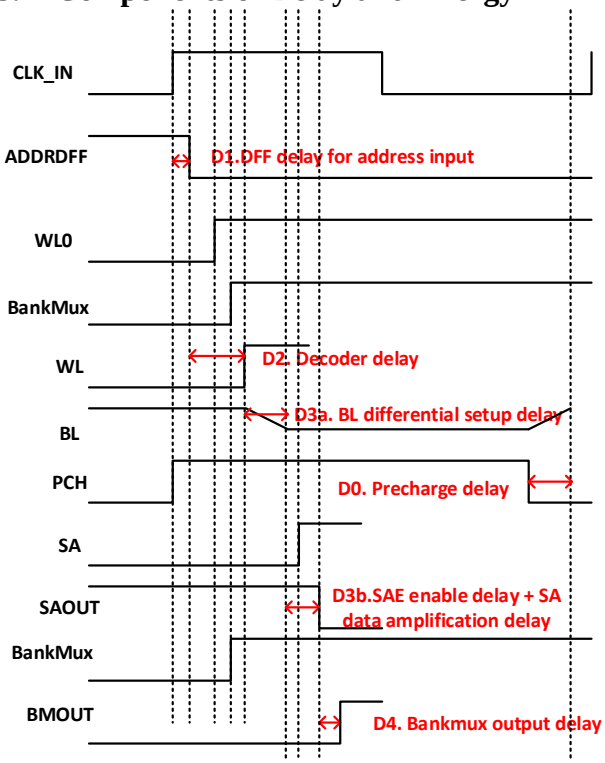

Figure 2. Timing diagram of read operation

To better understand the above research questions, a detailed register file model was carried out with showing delay of separated components [5]. Figure 2 shows the timing diagram of read operation. The total read delay was divided into five parts, which were D0 for precharge delay, D1 for D-flip-flop delay of input address, D2 for decoder delay, D3 (D3a and D3b) for BL differential setup delay and SA delay, D4 for bankmux delay. Total read delay could be calculated by the following equation:

Total delay= D1 + max(D2,D0)+(D3a+D3b)+D4

Figure 3 illustrates that decoder delay and bitline differential setup delay contributes the most to total read delay.

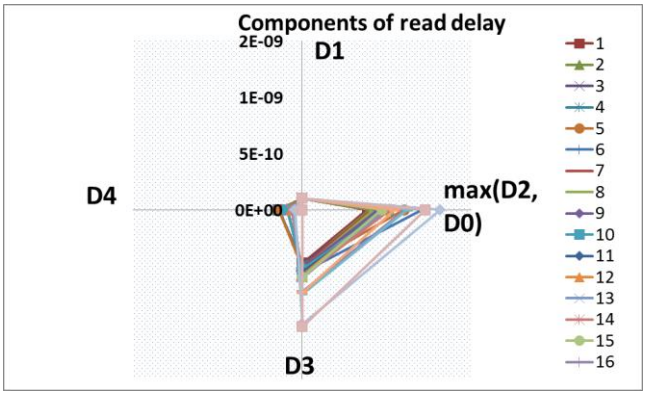

Figure 3 Components of read delay of 1KB memory capacity

In a similar way to read delay, the write delay was divided into five parts. The differences are D3 for input data delay, and D4 for write bitcell delay. According to Figure 4, total write delay is mainly constituted with the precharge delay, the decoder delay and the data input delay.

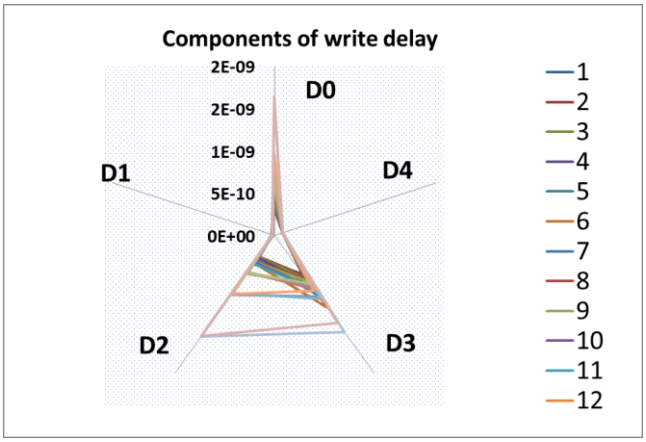

Figure 4 Components of write delay of 1KB memory capacity

3.2 Optimal Bank Structure of Different Memory Size

We simulated with ViPro for four different memory sizes, 0.25KB, 0.5KB, 1KB and 2KB. For each memory size, multiple bank prototypes were generated and plotted in Figure 5, from which we can see the distribution of delay and energy is

decreasing with the memory size getting smaller. Table 2 gives the optimized delay and optimized energy structures in the four memory sizes, and the NCol stands for the number of interleaving columns that sharing the data path. As we can see, in different memory size, the minimum delay occurs when the number of column mux is 1. This is because the WL capacitance is the smallest when column mux is 1, so the decoder delay is the smallest. For a bank, when column mux is 1 the load for precharge signal input is the smallest.

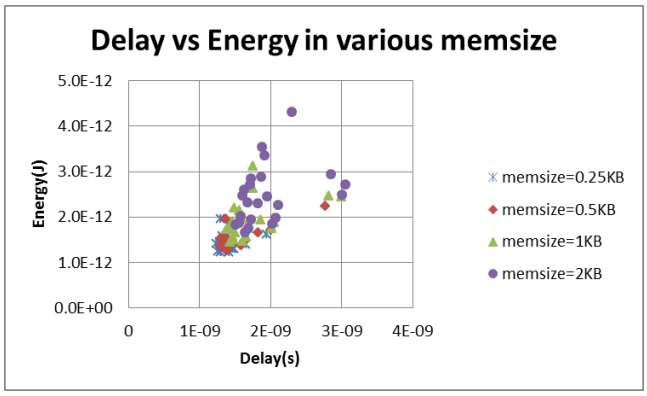

Figure 5 Relation of delay and energy in various memory sizes.

The minimum energy consumption occurs when the number of rows is around 64. This is because the energy spent on bitline of both read and write operation is higher when the number of rows are larger. Nevertheless, more energy is spent on bankmux and D-flip-flop when the number of rows is smaller (more banks required), because wire capacitance is more significant with more banks. To summarize, optimized delay and energy always occur when number of colmux is 1, if not taking area overhead into consideration; the most energy saving bank structure is when row number is around 64.

Table 2 Minimum delay and energy in various memory sizes.

memsize	NCol	NRow	NBank	Minimum Delay(ns)	Minimum Energy(pJ)
0.25KB	1	16	16	1.24	-
	1	64	4	-	1.22
0.5KB	1	32	16	1.30	-
	1	64	8	-	1.26
1KB	1	32	32	1.47	-
	1	64	16	-	1.54
2KB	1	64	32	1.51	-
	1	128	16	-	1.66

4. INFLUENCE OF SENSE AMPLIFIER ON REGISTER FILE

Sense amplifier (SA) is used in register file bank to accelerate speed of BLs developing voltage difference. In addition, a SA also help with reducing the total read energy by decreasing the extra power spent on selected and half selected bitcells in the same row. However, the cost of employing a SA is more area overhead and more design effort for dedicated timing constraints. Further, the benefits of utilizing a SA are becoming more insignificant for two reasons with the shrinking of VDD. The first reason is that resolution of SA requires a minimum voltage difference Vbl to be built on BLs, and the reduction of delay is becoming less effective when Vbl is a larger proportion of VDD. What's more, dynamic power consumed by SRAM is decided by 1/2*Vbl*Cbl*VDD, which is linearly reduced with VDD shrinking. The second reason is severe variation in recent technologies. Supported by the research of [6], variation worsens the access time dramatically in low voltage, because the delay distribution of BL is much wider in low voltages. As a consequence, faster BLs should wait for the worst case BL so that the whole array can function properly, and extra energy is being consumed. This paper also conveys the idea that SA is not needed in low voltage applications, as a result that the performance improvements of employing a SA is taken away by the worst case delay.

ViPro can facilitate in researching the difference of having or having not a SA by running simulation with two different types of register file bitcell. Both the bitcells in Figure 6 have 1 read port and 1 write port; (a) has two read BLs which can be used with a SA, while (b) only has a single read BL.

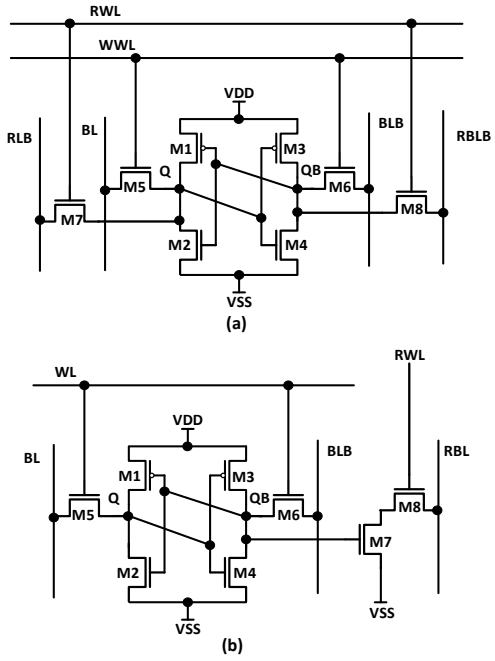

Figure 6 (a) Bitcell with SA (b) Bitcell without SA

4.1 Simulation Results of Large Memory Size

For larger memory size of 2KB, Figure 7 shows the trend of read energy and delay with the number of row and column mux with and without SA. The read energy reduction is remarkable with a SA when number of column mux is larger, as the maximum difference of two lines always occurs at the peak in (a). The read delay reduction is more significant when number of row is larger, as shown in (b).

Figure 8 shows that the optimized bank structure with a SA has much smaller energy and delay compared to without one.

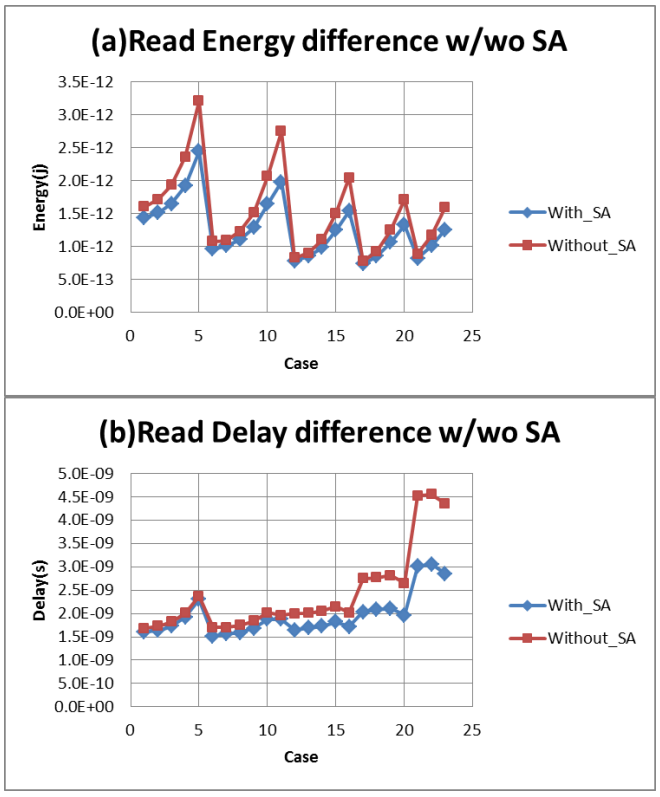

Figure 7. For memory size of 2KB, (a) Read energy difference with and without SA (b) Read delay difference with and without SA. From case 1 to case 23, the number of row grows from 32 to 512, and the number of column mux varies within 1 to 32.

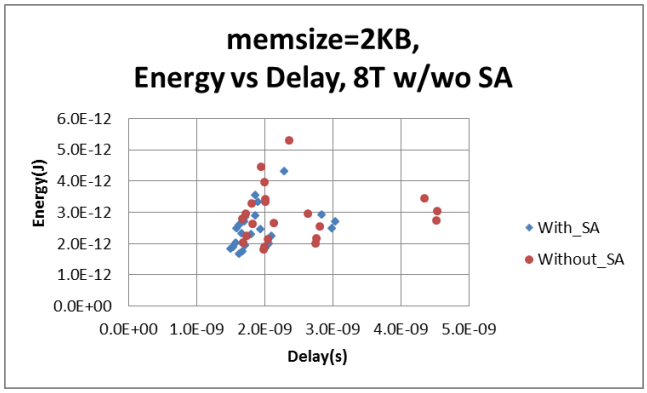

Figure 8. Relation of energy and delay with and without sense amplifier (SA) when memory size is 2KB

4.2 Simulation Results of Small Memory Size

As a comparison experiment to large memory size, we proposed another set of simulation with 0.25KB capacity. Figure 9 (a) and (b) show that the influence of utilizing SA is weaken across various bank structures when memory size is small. Even though the benefit of SA is remarkable in reducing the worst delay of bank with large row number and column mux number, the optimal energy and optimal delay are barely being improved, as shown in Figure 10.

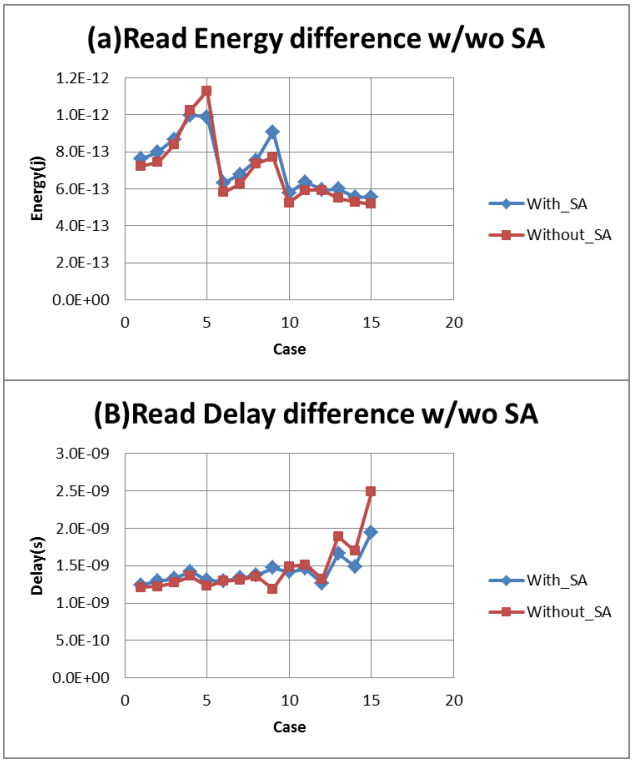

Figure 9 For memory size of 2KB, (a) Read energy difference with and without SA (b) Read delay difference with and without SA. From case 1 to case 15, the number of row grows from 16 to 256, and the number of column mux varies within 1 to 16.

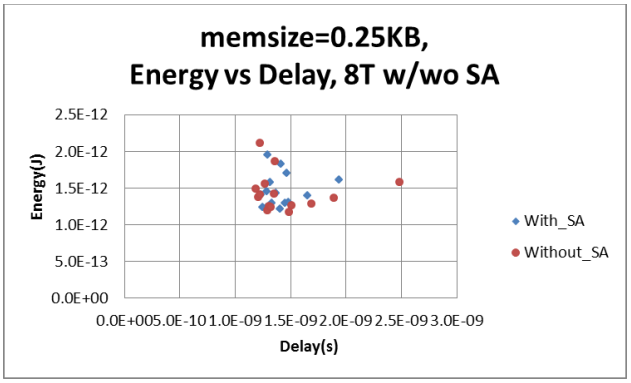

Figure 10 Relation of energy and delay with and without sense amplifier (SA) when memory size is 0.25KB

4.3 Conclusion

We filled the gap between 0.25KB and 2KB with two more set of experiments, in which the memory sizes are 0.5KB and 1KB. The effect of SA on optimal energy and optimal delay is more significant in large memory capacity. This is because SA can accelerate BL differential setup time, which is a notable part of total read delay when number of row is larger according to the results from part 3.1. Figure 11 shows the trend of energy and

delay improvement by utilizing a sense amplifier, which confirms our conclusion.

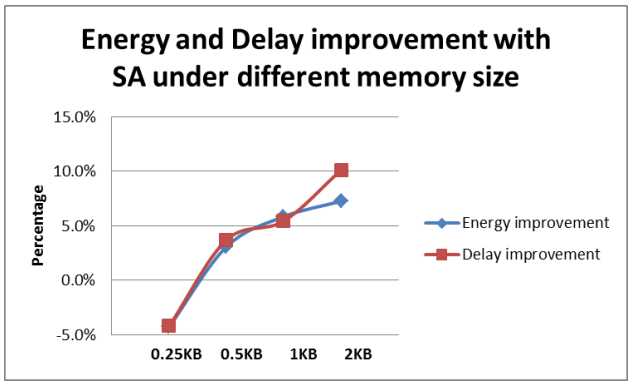

Figure 11 Energy and delay improvement with sense amplifier when memory sizes are 0.25KB, 0.5KB, 1KB and 2KB.

5. ACKNOWLEDGMENTS

Thanks to Prof. Benton Calhoun for giving us a whole semester of wonderful and enthusiastic lectures. Also, thanks to our TA Farah Yahya for patiently help with learning the VLSI.

6. REFERENCES

[1] Rakesh Nalluri and etc. Customization of Register File Banking Architecture for Low Power. VLSID 2007.

[2] Eric Donkoh and etc. A hybrid and Adaptive Model for Predicting Register File and SRAM Power Using a Reference Design. DAC 2012, June 3-7, P62-67.

[3] Benton Calhoun. VIRTUAL PROTOTYPING (VIPRO) TOOL FOR MEMORY SUBSYSTEM DESIGN EXPLORATION AND OPTIMIZATION.

[4] S. Nalam, M. Bhargava, K. Mai, B.H. Calhoun, "Virtual Prototyper (ViPro): an early design space exploration and optimization tool for SRAM designers," DAC, pp. 138-143, 2010.

[5] Koichi Takeda. Multi-Step Word-Line Control Technology in Hierarchical Cell Architecture for Scaled-Down High-Density SRAMs. IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 46, NO. 4, APRIL 2011.

[6] Energy Efficiency Degradation Caused by Random Variation in Low-Voltage SRAM and 26% Energy Reduction by Bitline Amplitude Limiting (BAL) Scheme. IEEE Asian Solid-State Circuits Conference, November 14-16, 2011 / Jeju, Korea.

[7] Kevin Linger. Automating the Design of Register Files in VLSI Chips: A Prototyping Tool.

[8] Xuan Guan and etc. Reducing Power Consumption of Embedded Processors through Register File Partitioning and Compiler Support. ASAP 2008, P269-274.

[9] Ataur R. Patwary. Bit-Line Organization in Register Files for Low Power AND HIGH-PERFORMANCE APPLICATIONS. ICECE 2006, P505-508.

[10] Ting-Sheng Jau and etc. Analysis and Design of High Performance, Low Power Multiple Ports Register Files. APCCAS 2006, P1453-1456.

[11] Hao Yan and etc. A Low-power 8-Read 4-Write Register File Design. APCPR 2010. Page 178-181.